ELF WP2 Modeling Guidelines

This document gives a short introduction to modelling of application schemas for the ELF specifications. The application schemas shall build upon the INSPIRE specifications and should therefore adhere to the requirements and recommendations in the INSPIRE Generic Conceptual Model (GCM).

Presentation to: WP2 – Webinar (WP2 Public Consultation)

Author: Morten Borrebaek

Date: 2014-11-20
https://service.projectplace.com/pp/pp.cgi/r934779671
Scope – important policy statement

The revised modelling guidelines in this document are formed in such a way that an existing INSPIRE implementation by default is conformant to an ELF specifications for the themes that are in the remit of the ELF data specifications. The impact of such a precondition is that all ELF additions have to be optional (not even voidable), and that there should be no constraints on the INSPIRE that affects the INSPIRE GML application schema.
Clause 2 - ELF modelling approach

Model structure / UML repository

Project Browser

Tree view of model structure:
- Model
  - ELF Model
    - ELF
      - Overall package model
        - ELFDataSpecification
          - Overview
            - applicationSchema Addresses
            - applicationSchema AdministrativeUnits
            - applicationSchema Buildings
            - applicationSchema CadastralParcels
            - Elevation
            - applicationSchema GeographicalNames
            - applicationSchema Hydrogeology
            - Hydrography
            - applicationSchema LandCover
            - applicationSchema Miscellaneous
            - applicationSchema ProtectedSites
            - applicationSchema SeaRegions
            - Statistical Units
            - Transport Networks
        - ELFProductSpecification
    - UseCases
    - INSPIRE Consolidated UML Model
ELF UML modelling Principles

To achieve INSPIRE compliancy in data modelling the following principles should be observed:

Extensions shall not

• Change the specification but normatively reference it with all its requirements
• Set any additional requirements that break any requirement of the INSPIRE data specification
• Add concepts that overlap with existing INSPIRE concepts
• Make a pure INSPIRE implementation non-conformant to the ELF specifications

Extensions may

• Add new application schemas importing INSPIRE or other schemas as needed
• Add new types and constraints in the new application schemas
• extend INSPIRE code lists if not centrally managed
Clause 2 – UML modelling principles

ELF UML modelling Principles (cont)

For each matching concept of INSPIRE and ELF identified from the analysis of the matching tables, a corresponding concept is created in ELF. Where possible, these concepts should be sub-classes of existing INSPIRE concepts (feature or data type, code lists etc.) by:

- Define additional optional attributes that are present in existing data or required by users but missing in INSPIRE
- Add constraints to ensure that ELF meets the user requirements where applicable
- Define new ELF feature types for concepts that are present in existing data or required by users but missing in INSPIRE
- For code lists in INSPIRE, identify matching codes and define additional codes where missing – reuse as many values from INSPIRE as possible and define new values only if no existing value can be matched. Describe any additional constraints (e.g. sometimes a code list value cannot be mapped or a code list value depends based on the value of another property).
- Add optional associations where required
- Avoid the stereotype <voidable> for new attributes and associations, to ensure that a 'pure' INSPIRE implementation conforms to ELF.
Clause 2 – UML modelling principles

Example: NamedPlace

1. Subtype all INSPIRE feature types valid for ELF (topographic reference data)
2. Define additional attributes
3. Add constraints
4. Define new ELF classes (feature types, datatypes, codelists)
5. Associate feature types (not in figure)
Step 1 - Subtyping INSPIRE feature types

++/administeredBy
 «voidable» 0..*

+upperLevelUnit
 «voidable» 0..1

+lowerLevelUnit
 «voidable» 0..*

+coAdminister
 «voidable» 0..*

«featureType»
AdministrativeUnit

(from ELF Model::INSPIRE Consolidated UML Model::
Themes::Annex t::Administrative Units::AdministrativeUnits)
Clause 2 – UML modelling principles

Step 2 - Adding new attributes to ELF feature types

(from ELF Model::INPIRE Consolidated UML Model::Themes::Annex I::Administrative Units::AdministrativeUnits)
Clause 2 – UML modelling principles

Step 3 - Adding constraints

```
Class : NamedPlace
    populationNumber : PopulationRange [0..1]
    constraints
        {geometryIsPointOrSurface}
        {populationNumberPopulatedPlace}
```

```
/* populationNumber must be present if the type of the NamedPlace is populatedPlace. */
inv: self.type->forAll(t | t = NamedPlaceTypeValue:populatedPlace) implies populationNumber->notEmpty()
```
Clause 2 – UML modelling principles

Step 4 and 5 - Define new ELF feature type and associate it to another ELF feature type

Adding a new feature type AdministrativeUnitArea associated to ELF administrativeUnit

SHNCode:
- identifier = country + nationalCode (default)
- identifierScheme = 'EuroBoundaryMap'

«featureType» AdministrativeUnit
+ country :CountryCode
+ geometry :GM_MultiSurface
+ inspire :Identifier
+ name :GeographicalName [1..*]
+ nationalCode :CharacterString
+ nationalLevel :AdministrativeHierarchyLevel

«voidable, lifecycleInfo»
+ beginLifespanVersion :DateTime
+ endLifespanVersion :DateTime [0..1]

«voidable»
+ nationalLevelName :LocalisedCharacterString [1..*]
+ residenceOfAuthority :ResidenceOfAuthority [1..*]

constraints
{CondominiumsAtCountryLevel}
{AdministrativeUnitHighestLevel}
{AdministrativeUnitLowestLevel}

(from ELF Model::INSPIRE Consolidated UML Model::Themes::Annex I::Administrative Units::AdministrativeUnits)

«featureType» AdministrativeUnitArea
+ geometry :GM_Surface
+ inspireID :Identifier
+ landCoverType :LandCoverTypeValue

«voidable»
+ beginLifespanVersion :DateTime
+ endLifespanVersion :DateTime [0..1]
All additional ELF classifiers shall contain definitions sufficient for understanding of all classes, attributes, associations, operations and appropriate data type definitions.
Clause 3 – Documentation - diagrams

Context diagrams
**ELF UML profile (technicalities)**

The modelling guidelines specifies a profile of UML to be used in ELF, based upon similar UML profile for INSPIRE.

The ELF UML profile adds some additional tagged values to the INSPIRE UML profile. Different name spaces are defined, in addition to the xsdEncodingRule that is specific to ELF.

### Implementation

https://service.projectplace.com/pp/pp.cgi/r1025631019

---

<table>
<thead>
<tr>
<th>Stereotype</th>
<th>Tagged Value</th>
<th>Description</th>
<th>Remark</th>
<th>Value and examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>suppress</td>
<td></td>
<td>When true, identifies the feature type as a type that is created only in the ELF UML application schema to indicate that (1) the INSPIRE feature type is included in the ELF data specification and/or (2) to attach a constraint. When true, this feature type will be suppressed in the GML application schema.</td>
<td></td>
<td>Example: <a href="http://www.locationframework.eu/schemas/LandCover/1.0">http://www.locationframework.eu/schemas/LandCover/1.0</a></td>
</tr>
<tr>
<td>profiles</td>
<td></td>
<td>Comma-separated list of profile indicators which associates this model element to one or more levels of detail.</td>
<td></td>
<td>This structure allows one to have absolute and path which can be used.</td>
</tr>
<tr>
<td>dataType</td>
<td>xsdEncodingRule</td>
<td>XML Schema encoding rule to apply</td>
<td>Mandatory</td>
<td>MasterLoD0, MasterLoD1, MasterLoD2, Regional, Global. If empty, it applies to all LoD's.</td>
</tr>
</tbody>
</table>

---

**EUROPEAN LOCATION FRAMEWORK**

the Competitiveness and Innovation framework Programme (CIP)
ICT Policy Support Programme (PSP) Call 6 (Grant 325140)
## Colour scheme

<table>
<thead>
<tr>
<th>Overview/Feature Types &amp; data types</th>
<th>Apply color coding system for Feature Types and data types</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- INSPIRE white</td>
</tr>
<tr>
<td></td>
<td>- ELF light green</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>codelists</th>
<th>Apply color coding system for codelists</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- INSPIRE codelists light yellow</td>
</tr>
<tr>
<td></td>
<td>- ELF additional codelists yellow</td>
</tr>
</tbody>
</table>
Configuration files for the generation of GML application schemas and feature catalogues will be made available by request.

Questions?